

Testing Unateness of Real-Valued Functions*

Roksana Baleshzar[†] Meiram Murzabulatov[†] Ramesh Krishnan S. Pallavoor[†]

Sofya Raskhodnikova[†]

Abstract

We give a unateness tester for functions of the form $f : [n]^d \rightarrow R$, where $n, d \in \mathbb{N}$ and $R \subseteq \mathbb{R}$ with query complexity $O(\frac{d \log(\max(d, n))}{\epsilon})$. Previously known unateness testers work only for Boolean functions over the domain $\{0, 1\}^d$. We show that every unateness tester for real-valued functions over hypergrid has query complexity $\Omega(\min\{d, |R|^2\})$. Consequently, our tester is nearly optimal for real-valued functions over $\{0, 1\}^d$. We also prove that every nonadaptive, 1-sided error unateness tester for Boolean functions needs $\Omega(\sqrt{d}/\epsilon)$ queries. Previously, no lower bounds for testing unateness were known.

1 Introduction

We study property testing unateness of functions of the form $f : [n]^d \rightarrow R$, where $n, d \in \mathbb{N}$ and $R \subseteq \mathbb{R}$. Unate functions are used in mathematics and other technical disciplines, for example in tautology checking [14, 15] and in switching theory [17]. For $x, y \in [n]^d$, where $x = x_1, \dots, x_d$ and $y = y_1, \dots, y_d$, we define $|x - y|_1 = \sum_{i=1}^n |x_i - y_i|$. Two points x, y are **neighbours** if $|x - y|_1 = 1$. A function $f : [n]^d \rightarrow \mathbb{R}$ is **unate** if each dimension $i \in [d]$ can be assigned an associated direction **up** or **down**; the direction is **up** if $f(x) \leq f(y)$ for all neighbours x, y where $y_i = x_i + 1$; and the direction is **down** if $f(x) \geq f(y)$ for all such x and y . Moreover, a function is **monotone** if the direction is **up** in all dimensions. Thus, unateness is a generalization of monotonicity.

The domain $[n]^d$ is called a **hypergrid** and the special case $\{0, 1\}^d$ is called a **hypercube**. The distance between two functions $f, g : [n]^d \rightarrow \mathbb{R}$ is equal to the fraction of points $x \in [n]^d$ where $f(x) \neq g(x)$. Given a parameter $\epsilon \in (0, 1)$, two functions f and g are ϵ -far from each other if the distance between f and g is at least ϵ . A function f is ϵ -far from a property P if it is ϵ -far from any function which has property P . A **property tester** [10, 16] for a property P is a randomized algorithm which, given parameter $\epsilon \in (0, 1)$ and oracle access to the input function f , accepts f with probability $\frac{2}{3}$, if it has the property P , and rejects f with probability $\frac{2}{3}$, if it is ϵ -far from P . A tester is nonadaptive if it makes all queries in advance, and adaptive otherwise. A testing algorithm for property P has 1-sided error if it always accepts all inputs that satisfy P and 2-sided error, otherwise. A **unateness tester** is a property tester for unateness. To the best of our knowledge, testing unateness has been studied previously only for Boolean functions over the hypercube domain.

*This work was partially supported by NSF award CCF-1422975.

[†]Pennsylvania State University, rxb5410@cse.psu.edu, mzm269@psu.edu, rxp271@cse.psu.edu, sofya@cse.psu.edu.

1.1 Our results

In this paper, we give a unateness tester for real-valued function over the hypergrid. It is the first unateness tester for this type of functions. We also present first lower bounds on the number of queries required to test unateness. In particular, our tester and lower bound differ only by a logarithmic factor for real-valued functions over the hypercube.

The query complexity of our unateness tester is $O(\frac{d \log(\max(d, n))}{\epsilon})$ for functions $f : [n]^d \rightarrow \mathbb{R}$. Our tester generalizes the tester of Khot and Shinkar [12] to real-valued functions over the hypergrid domain. Their tester has query complexity $O(\frac{d \log d}{\epsilon})$ and it works for Boolean functions over the hypercube domain.

We also present two lower bounds for testing unateness of functions over the hypercube domain. The first lower bound of $\Omega(\min\{d, |R|^2\})$, where R is the range of the function, is over the real-valued functions. Our second lower bound of $\Omega(\sqrt{d}/\epsilon)$ is for nonadaptive, 1-sided error unateness testers over Boolean functions. Our lower bounds build on the same lower bounds for monotonicity testing by Blais et al. [3] and Fischer et al. [9] of Boolean functions over the hypercube domain.

1.2 Previous work

The problem of testing unateness was first considered by Goldreich et al. [11]. In their work, by extending their monotonicity tester, they obtain a nonadaptive, 1-sided error tester for unateness with query complexity $O(\frac{d^{3/2}}{\epsilon})$. Recently, Khot and Shinkar [12] gave an adaptive unateness tester with query complexity $O(\frac{d \log d}{\epsilon})$.

The related property of monotonicity has been studied extensively for different types of functions in the context of property testing. The problem of testing monotonicity of Boolean functions over the hypercube domain was first introduced by Goldreich et al. [11]. It was shown in [8, 11] that monotonicity for hypercube domain can be tested with query complexity $O(\frac{d}{\epsilon})$. A monotonicity tester with better query complexity of $\tilde{O}(\frac{d^{7/8}}{\epsilon^{3/2}})$ was introduced by Chakrabarty and Seshadhri [5]. Chen et al. [6] modified the tester of Chakrabarty and Seshadhri [5] and improved the query complexity to $\tilde{O}(\frac{d^{5/6}}{\epsilon^4})$. Most recently, Khot et al. [13] improved the query complexity of the tester to $\tilde{O}(\frac{\sqrt{d}}{\epsilon^2})$.

The lower bound for any nonadaptive one-sided error tester for monotonicity over the hypercube was proved to be $\Omega(\sqrt{d})$ by Fischer et al. [9]. Also, Chen et al. [7] gave a lower bound of almost $\Omega(\sqrt{d})$ for any nonadaptive, two-sided error tester. Moreover, there is a lower bound of $\Omega(\min\{d, |R|^2\})$ over the real-valued functions by Blais et al. [3]. Most recently, Belovs and Blais [1] gave a lower bound of $\tilde{\Omega}(d^{1/4})$ for adaptive testers of Boolean functions over the hypercube. Chakrabarty and Seshadhri [4] proved that any adaptive, two-sided monotonicity tester for functions $f : [n]^d \rightarrow \mathbb{N}$ must make $\Omega(\frac{d \log n - \log \epsilon^{-1}}{\epsilon})$ queries.

2 Algorithm for Testing Unateness

In this section, we present an algorithm to test unateness of real-valued functions over the hypergrid domain and prove Theorem 2.1.

We use the notation $z \circ_T w$ to denote concatenation with respect to the locations in the set T . More specifically, this operation fills the locations in T with $|T|$ bits in z , and the other $[d] \setminus T$ locations with $d - |T|$ bits in w , respectively.

Theorem 2.1. *For a given function $f : [n]^d \rightarrow \mathbb{R}$, there exists a 1-sided error unateness tester with*

a query complexity of $O(\frac{d \log(\max(d, n))}{\epsilon})$, where $\epsilon \in (0, \frac{1}{2})$ is the proximity parameter.

Proof. The main idea of the tester described in Algorithm 1 is as follows. Given a function $f : [n]^d \rightarrow \mathbb{R}$, the tester finds a subset of influential dimensions $T \subseteq [d]$, such that the dependence of f on the remaining dimensions, $[d] \setminus T$, is negligible. Furthermore, for each influential dimension $i \in T$, the tester finds an edge $(x, x + e_i)$, where $f(x) \neq f(x + e_i)$, and assigns a direction to the i -th dimension according to the value of f at these two points. The direction is **up** if $f(x) < f(x + e_i)$ and it is **down** if $f(x) > f(x + e_i)$. After that, the algorithm applies a monotonicity tester on f with respect to the directions for the dimensions in T and some random assignments w for other dimensions, and outputs the answer.

Algorithm 1 Unateness Tester($f : [n]^d \rightarrow \mathbb{R}, \epsilon$)

- 1: Let $m = O(\frac{d}{\epsilon})$ and $T = \emptyset$
 - 2: **for** $i = 1 \dots m$ **do**
 - 3: Find_an_Influential_Dimension(f, T)
 - 4: **if** it returns a dimension and its direction (i^*, b_{i^*}) **then**
 - 5: Add i^* to T , and let b_{i^*} be its direction
 - 6: **end if**
 - 7: **end for**
 - 8: Pick a uniform $w \in [n]^{[d] \setminus T}$
 - 9: Let $f_w(z) = f(z \circ_T w)$ for all $z \in [n]^T$
 - 10: Apply the monotonicity tester by Chakrabarty and Seshadhri [4] on f_w with respect to the directions $\{b_i : i \in T\}$ and proximity parameter $\frac{\epsilon}{2}$
 - 11: **return** the output of the monotonicity tester
-

In order to find influential dimensions, the tester uses the function Find_an_Influential_Dimension, which is presented in Algorithm 2. It works as a subroutine and tries to find an influential dimension in each run. It gets oracle access to a function $f : [n]^d \rightarrow \mathbb{R}$, and a subset of dimensions $T \subseteq [d]$, which is empty at the beginning. This subroutine outputs \perp if it doesn't find any new influential dimension. Otherwise, it outputs a dimension $i^* \in [d] \setminus T$ and its direction $b_{i^*} \in \{\text{up}, \text{down}\}$.

This subroutine can find a new influential dimension and its direction with high probability if f has non-negligible dependence on the dimensions in $[d] \setminus T$. Each time we run Find_an_Influential_Dimension, it picks two points $x, y \in [n]^d$ uniformly and independently at random, whose coordinates are equal in the dimensions in T . If $f(x) \neq f(y)$, it uses **binary search** to decrease the distance between x and y to 1, while ensuring that $f(x) \neq f(y)$ still holds.

If the input function f is unate, then the tester always accepts, as Algorithm 2 always finds the correct direction for each influential dimension and the function with respect to the influential dimensions is monotone.

In order to show the correctness of Algorithm 1, we will prove the following two lemmas. In these lemmas let r denote the number of different values in the image of the function f , and P_k^z where $k \in [r]$ and $z \in [n]^T$ be the fraction of the k^{th} frequent value among the values of $f(x|_{x_T=z})$.

Lemma 2.2. *Let $f : [n]^d \rightarrow \mathbb{R}$ and $T \subseteq [d]$ be the set in Algorithm 1 after running $m = O(\frac{d}{\epsilon})$ iterations of Algorithm 2. Then, with probability at least $5/6$, the set T satisfies*

$$\mathbb{E}_{z \in [n]^T} \left[\sum_{k=1}^r P_k^z (1 - P_k^z) \right] < \frac{\epsilon}{16}.$$

Algorithm 2 Find_an_Influential_Dimension($f : [n]^d \rightarrow \mathbb{R}, T$)

```

1: Pick  $x, y \in_R [n]^d$  independently and uniformly at random such that  $x_T = y_T$ 
2: if  $f(x) = f(y)$  then
3:   return  $\perp$ 
4: else
5:   do
6:      $U \leftarrow \{i \in [d] : x_i = y_i\}$ 
7:      $V \leftarrow \{i \in [d] : x_i \neq y_i\}$ 
8:     Let  $V = \{v_1, \dots, v_t\}$ 
9:     Set  $z_V = x_{v_1} \circ \dots \circ x_{v_{\lfloor \frac{t}{2} \rfloor}} \circ y_{v_{\lfloor \frac{t}{2} \rfloor + 1}} \circ \dots \circ y_{v_t}$ 
10:    Let  $z = x_U \circ_T z_V \in [n]^d$ 
11:    if  $f(x) \neq f(z)$  then
12:       $y \leftarrow z$ 
13:    else
14:       $x \leftarrow z$ 
15:    end if
16:    while  $|V| \neq 1$ 
17:      Let  $i^* \in [d]$  be the element in  $V$ 
18:      Let  $b \in \{up, down\}$  be the direction of edge  $(x, y)$ 
19:      return  $(i^*, b)$ 
20: end if

```

Lemma 2.3. Let $f : [n]^d \rightarrow \mathbb{R}$ be a real-valued function, and let $T \subseteq [d]$ such that,

$$\mathbb{E}_{z \in [n]^T} \left[\sum_{k=1}^r P_k^z (1 - P_k^z) \right] < \frac{\epsilon}{16}. \quad (1)$$

Then, for a random $w \in [n]^{[d] \setminus T}$,

$$\Pr_{w \in [n]^{[d] \setminus T}} [\text{dist}(f_w, f) > \frac{\epsilon}{2}] \leq \frac{1}{6}.$$

Proof of Lemma 2.2. First, we show that $\mathbb{E}_{z \in [n]^T} \left[\sum_{k=1}^r P_k^z (1 - P_k^z) \right]$ is equal to the probability that Algorithm 2 finds a new dimension on input (f, T) . Algorithm 2 finds a new dimension if and only if $f(x) \neq f(y)$ in Line 2. By definition of P_k^z , the probability of choosing x such that $f(x) = k$ is P_k^z , and the probability of choosing y such that $f(y) \neq k$ is $1 - P_k^z$. The total probability of choosing two different values is $\sum_{k=1}^r P_k^z (1 - P_k^z)$ for a fixed z . So, the probability over all possible z 's is $\mathbb{E}_{z \in [n]^T} \left[\sum_{k=1}^r P_k^z (1 - P_k^z) \right]$.

Now, we state the following claim to prove that when the Algorithm 2 finds a new dimension then the value of $\mathbb{E}_{z \in [n]^T} \left[\sum_{k=1}^r P_k^z (1 - P_k^z) \right]$ does not increase.

Claim 2.4. For $\forall T \subseteq [d]$ and $\forall i \in [d]$, when the dimension i is added to the set T , the value of

$$\mathbb{E}_{z \in [n]^T} \left[\sum_{k=1}^r P_k^z (1 - P_k^z) \right]$$

does not increase.

Proof. Let $P_{k,T}^z$ denote P_k^z when $z \in [n]^T$. Let $T' = T \cup \{i\}$. Define z_j , to be equal to z for all dimensions in T , and equal to j in the i^{th} dimension, for each $j \in [n]$. To simplify notation for the rest of the proof, let $P = P_{k,T}^z$, $P^j = P_{k,T'}^{z_j}$, for $j \in [n]$. Then, $P = \sum_{j=1}^n \frac{P^j}{n}$. The probability that Algorithm 2 finds a new dimension on input (f, T') is

$$\mathbb{E}_{z \in [n]^T} \left[\sum_{k=1}^r \sum_{j=1}^n \frac{P^j (1 - P^j)}{n} \right].$$

To complete the proof, it is enough to show that $g(P) \geq \sum_{j=1}^n \frac{g(P^j)}{n}$, where $g(x) = x(1 - x)$. This inequality holds because $P = \sum_{j=1}^n \frac{P^j}{n}$ and g is concave. \square

The following claim completes the proof of Lemma 2.2.

Claim 2.5. After $m = O(\frac{d}{\epsilon})$ iterations of Algorithm 2, either the size of the set T is equal to d , or with probability at least $5/6$, the following inequality holds,

$$\mathbb{E}_{z \in [n]^T} \left[\sum_{k=1}^r P_k^z (1 - P_k^z) \right] < \frac{\epsilon}{16}.$$

Proof. Let Y_i be an indicator random variable, which is equal to 1 if Algorithm 2 finds a new dimension in the i^{th} iteration and 0, otherwise. We have $\Pr[Y_i] = \mathbb{E}_{z \in [n]^{T_i}} [Y_i] = \mathbb{E}_{z \in [n]^{T_i}} \left[\sum_{k=1}^r P_k^z (1 - P_k^z) \right]$, where T_i is the set of influential dimensions in the i^{th} iteration. Hence, the total number of dimensions that Algorithm 2 finds after m iterations is $Y = \sum_{i=1}^m Y_i$. Now, after m iterations, if the following holds:

$$\mathbb{E}_{z \in [n]^T} \left[\sum_{k=1}^r P_k^z (1 - P_k^z) \right] \geq \frac{\epsilon}{16},$$

then,

$$\mathbb{E}_{z \in [n]^{|T|}} [Y] \geq \frac{m\epsilon}{16}.$$

Using Chernoff Bound we have,

$$\Pr_{z \in [n]^T} [Y \leq \frac{m\epsilon}{32}] \leq e^{-\frac{m\epsilon}{128}}.$$

For $m = \frac{32d}{\epsilon}$ and $d \geq 8$, we have

$$\Pr_{z \in [n]^T} [Y \leq d] \leq e^{-\frac{d}{4}} \leq \frac{1}{6}.$$

□

This completes the proof of Lemma 2.2. □

Proof of Lemma 2.3. Define the plurality function $\mathbf{Pl}_T : [n]^T \rightarrow [r]$ as $\mathbf{Pl}_T(z) = k$ such that P_k^z is the maximum (ties are broken arbitrarily).

By assumption (1) for a uniformly random $w \in [n]^{[d] \setminus T}$, it holds that

$$\begin{aligned} \mathbb{E}_{w \in [n]^{[d] \setminus T}}[\mathbf{dist}(f_w, \mathbf{Pl}_T)] &= \mathbb{E}_{w \in [n]^{[d] \setminus T}} \left[\Pr_{z \in [n]^T} [f_w(z) \neq \mathbf{Pl}_T(z)] \right] \\ &= \mathbb{E}_{w \in [n]^{[d] \setminus T}} \left[\mathbb{E}_{z \in [n]^T} [\mathbb{1}(f_w(z) \neq \mathbf{Pl}_T(z))] \right] \\ &= \mathbb{E}_{z \in [n]^T} \left[\mathbb{E}_{w \in [n]^{[d] \setminus T}} [\mathbb{1}(f_w(z) \neq \mathbf{Pl}_T(z))] \right] \\ &= \mathbb{E}_{z \in [n]^T} \left[\Pr_{w \in [n]^{[d] \setminus T}} [f_w(z) \neq \mathbf{Pl}_T(z)] \right]. \end{aligned}$$

Without loss of generality, assume that P_1^z is the probability of the value with the highest frequency in $f(x|_{x_T=z})$. Hence,

$$\mathbb{E}_{z \in [n]^T} \left[\Pr_{w \in [n]^{[d] \setminus T}} [f_w(z) \neq \mathbf{Pl}_T(z)] \right] = \mathbb{E}_{z \in [n]^T} [1 - P_1^z].$$

Claim 2.6. For all $z \in [n]^T$,

$$1 - P_1^z \leq \sum_{k=1}^r P_k^z (1 - P_k^z).$$

Proof. To simplify notations for the rest of the proof, let $P_k = P_k^z$. We have

$$\sum_{k=1}^r P_k (1 - P_k) = 1 - \sum_{k=1}^r P_k^2.$$

So it is enough to show that, $1 - P_1 \leq 1 - \sum_{k=1}^r P_k^2$ or $\sum_{k=1}^r P_k^2 \leq P_1$ which can be proved as follows,

$$\sum_{k=1}^r \frac{P_k^2}{P_1} \leq \sum_{k=1}^r \frac{P_k^2}{P_k} = 1$$

□

By Claim 2.6 and our initial assumption (1), we have

$$\mathbb{E}_{z \in [n]^T} [1 - P_1^z] \leq \mathbb{E}_{z \in [n]^T} \left[\sum_{k=1}^r P_k^z (1 - P_k^z) \right] < \frac{\epsilon}{16}.$$

So,

$$\mathbb{E}_{w \in [n]^{[d] \setminus T}} [\mathbf{dist}(f_w, \mathbf{Pl}_T)] < \frac{\epsilon}{16}.$$

By Markov's inequality,

$$\Pr_{w \in [n]^{[d] \setminus T}} \left[\text{dist}(f_w, \text{Pl}_T) \geq \frac{3\epsilon}{8} \right] \leq \frac{1}{6}.$$

Moreover,

$$\text{dist}(f, \text{Pl}_T) = \mathbb{E}_{w \in [n]^{[d] \setminus T}} [\text{dist}(f_w, \text{Pl}_T)] < \frac{\epsilon}{16}.$$

Therefore, by triangle inequality we have,

$$\Pr_{w \in [n]^{[d] \setminus T}} \left[\text{dist}(f(x|_{x_{[d] \setminus T} = w}), f) \geq \frac{3\epsilon}{8} + \frac{\epsilon}{16} \right] \leq \Pr_{w \in [n]^{[d] \setminus T}} [\text{dist}(f_w, \text{Pl}_T) \geq \frac{7\epsilon}{16}] \leq \frac{1}{6}.$$

This completes the proof of Lemma 2.3. \square

The proof of Theorem 2.1 follows from Lemmas 2.2 and 2.3. Let $T \subseteq [d]$ be the set of influential dimensions in the unateness tester after $m = O(\frac{d}{\epsilon})$ iterations of Algorithm 2. By Lemma 2.2 the following inequality holds with probability at least $5/6$

$$\mathbb{E}_{z \in [n]^T} \left[\sum_{k=1}^r P_k^z (1 - P_k^z) \right] < \frac{\epsilon}{16}.$$

If T satisfies the above inequality and f is ϵ -far from being unate, then by Lemma 2.3, for a random $w \in [n]^{[d] \setminus T}$, the function $f(x|_{x_{[d] \setminus T} = w})$ is $\frac{\epsilon}{2}$ -far from being unate with probability at least $\frac{5}{6}$. And with the same probability it is $\frac{\epsilon}{2}$ -far from being monotone with respect to the directions of dimensions in the set T . So, with high probability the monotonicity tester in Algorithm 1 will reject $f(x|_{x_{[d] \setminus T} = w})$ with respect to the directions of the dimensions in T .

Now, we analyze the query complexity of Algorithm 1. Each call of Algorithm 2 makes $O(\log d)$ queries to perform the binary search. Algorithm 2 is called $O(\frac{d}{\epsilon})$ times by Algorithm 1. At the end, Algorithm 1 runs monotonicity tester with query complexity $O(\frac{d \log n}{\epsilon})$. So, the total query complexity of Algorithm 1 is $O(\frac{d \log(\max(d, n))}{\epsilon})$.

This completes the proof of Theorem 2.1. \square

3 Lower Bounds for Unateness

3.1 Lower Bound for Real-Valued Functions

In this section, we prove an adaptive, 2-sided error lower bound for testing unateness of functions over the hypergrid domain. This lower bound technique builds on the one used for proving the lower bound for monotonicity by Blais et al. [3].

Theorem 3.1. *Testing $f : \{0, 1\}^d \rightarrow R$ for unateness requires $\Omega(\min\{d, |R|^2\})$ queries.*

Proof. First, we consider the case when $R = \mathbb{R}$. We will prove the lower bound for testing unateness in this case by a reduction from Set-Disjointness problem.

Definition 3.1. (*Set-Disjointness*) *In the Set-Disjointness problem, there are two players Alice and Bob. Alice receives a string S as input and Bob receives a string T as input, where both strings have size k . Alice and Bob also have access to a random string. They must compute the value of function $\text{DISJ}_k(S, T)$, which is 1 if $S \cap T = \emptyset$ and 0 otherwise.*

Theorem 3.2. *Testing $f : \{0, 1\}^d \rightarrow \mathbb{R}$ for unateness requires $\Omega(d)$ queries.*

Proof. Let $S, T \subseteq [d]$ be the subsets received by Alice and Bob as the input to an instance of the Set-Disjointness problem. Alice and Bob build functions $\chi_S, \chi_T : \{0, 1\}^d \rightarrow \{-1, 1\}$, respectively, where $\chi_S(x) = (-1)^{\sum_{i \in S} x_i}$ and $\chi_T(x) = (-1)^{\sum_{i \in T} x_i}$. Let $h : \{0, 1\}^d \rightarrow \mathbb{Z}$ be a function defined by $h(x) = 2|x| + \chi_S(x) + \chi_T(x)$.

Claim 3.3. *The following holds:*

- (a) *The function h is unate if S and T are disjoint.*
- (b) *The function h is $\frac{1}{8}$ -far from unate, otherwise.*

Proof. Fix $i \in [d]$. For every $x \in \{0, 1\}^d$, we define $x^0, x^1 \in \{0, 1\}^d$ to be the vectors obtained by fixing the i^{th} coordinate of x to 0 and 1, respectively.

First, we prove part (a). Since S and T are disjoint, there are three cases:

- 1. $i \notin S$ and $i \notin T$. Then,

$$h(x^1) - h(x^0) = 2|x^1| - 2|x^0| = 2 > 0.$$

- 2. $i \in S$ and $i \notin T$. Then,

$$h(x^1) - h(x^0) = 2|x^1| - 2|x^0| - 2\chi_S(x^0) \geq 0.$$

- 3. $i \notin S$ and $i \in T$. Then,

$$h(x^1) - h(x^0) = 2|x^1| - 2|x^0| - 2\chi_T(x^0) \geq 0.$$

So, when S and T are disjoint, the function h is non-decreasing along dimension i .

For the part (b) of our claim, since $S \cap T \neq \emptyset$ there exists $i \in S \cap T$. For this i , we will show that there exists $\frac{1}{4}$ -fraction of pairs of the form (x^0, x^1) such that $h(x^1) < h(x^0)$ and another $\frac{1}{4}$ -fraction of pairs such that $h(x^1) > h(x^0)$ implying that h is not unate. We have

$$h(x^1) - h(x^0) = 2|x^1| - 2|x^0| - 2\chi_S(x^0) - 2\chi_T(x^0).$$

It is clear that if $\chi_S(x^0) = \chi_T(x^0) = 1$ then $h(x^1) < h(x^0)$. And if $\chi_S(x^0) = \chi_T(x^0) = -1$ then $h(x^1) > h(x^0)$. Suppose that $j \in S$ and also $j \neq i$. Let \bar{x}^0 be x^0 with the j^{th} bit flipped. Then, $\chi_S(x^0) = -\chi_S(\bar{x}^0)$, which means $\chi_S(x^0) = 1$ with probability $\frac{1}{2}$. By the same argument, $\chi_T(x^0) = 1$ with probability $\frac{1}{2}$. So, at least for $\frac{1}{4}$ of these pairs we have $\chi_S(x^0) = \chi_T(x^0) = 1$. By using the same technique we can prove that for at least $\frac{1}{4}$ fraction of these pairs we have $\chi_S(x^0) = \chi_T(x^0) = -1$. Therefore, when S and T are overlapping, h is $\frac{1}{8}$ -far from unate. \square

This completes the proof of Theorem 3.2 \square

Now consider the case when $|R| > 12\sqrt{d} + 5$. We can again use a reduction from set-disjointness problem.

Theorem 3.4. *Testing $f : \{0, 1\}^d \rightarrow R$ for unateness, where $|R| > 12\sqrt{d} + 5$, requires $\Omega(d)$ queries.*

Proof. Without loss of generality we can also assume that $\{d - 6\sqrt{d} - 2, \dots, d + 6\sqrt{d} + 2\} \subseteq R$. Let $\chi_S, \chi_T : \{0, 1\}^d \rightarrow \{-1, 1\}$ be the functions given to Alice and Bob in the communication game, where $S, T \subseteq [d]$. Now we will define the function $h' : \{0, 1\}^d \rightarrow \mathbb{Z}$ (as defined in [3]), as follows:

$$h'(x) = \begin{cases} d - 6\sqrt{d} - 2 & \text{if } |x| < \frac{d}{2} - 3\sqrt{d}; \\ d + 6\sqrt{d} + 2 & \text{if } |x| > \frac{d}{2} + 3\sqrt{d}; \\ h(x) & \text{otherwise.} \end{cases}$$

This function rounds $h(x)$ up to $d - 6\sqrt{d} - 2$ for strings with low Hamming distance and rounds $h(x)$ down to $d + 6\sqrt{d} + 2$ for strings with high Hamming distance. It also keeps the actual value of h for points x such that $\frac{d}{2} - 3\sqrt{d} \leq |x| \leq \frac{d}{2} + 3\sqrt{d}$. By the definition of $h(x)$, this value will be between $d - 6\sqrt{d} - 2$ and $d + 6\sqrt{d} + 2$.

Claim 3.5. *Function h' is unate if h is unate and h' is $1/16$ -far from unate when h is $1/8$ -far from unate.*

Proof. Since every monotone function is also unate, the first part of the claim directly follows from Blais et al. [3].

For the second part we prove that the distance between h and h' is at most $\frac{1}{16}$. Thus, when S and T intersect, h is $\frac{1}{8}$ -far from unate and so h' is $\frac{1}{16}$ -far from unate. Let x be a uniformly random point from $\{0, 1\}^d$. We know that $h'(x) = h(x)$ when $\frac{d}{2} - 3\sqrt{d} \leq |x| \leq \frac{d}{2} + 3\sqrt{d}$. By Chernoff Bound,

$$\Pr \left[\left| |x| - \frac{d}{2} \right| > 3\sqrt{d} \right] < 0.03 < \frac{1}{16} \Rightarrow \Pr[h'(x) \neq h(x)] < \frac{1}{16}.$$

By triangle inequality, h' is at least $\frac{1}{16}$ -far from unate. □

This completes the proof of Theorem 3.4. □

For the last case, we prove the following theorem.

Theorem 3.6. *Testing a function $f : \{0, 1\}^d \rightarrow R$ for unateness, where $|R| = o(\sqrt{d})$, requires $\Omega(|R|^2)$ queries.*

Proof. In this case, we can prove the lower bound by using a reduction to Theorem 3.4. Let m be the largest integer such that $|R| > 12\sqrt{m} + 5$. We prove the following claim.

Claim 3.7. *For every function $g : \{0, 1\}^m \rightarrow R$, there exists a function $h : \{0, 1\}^d \rightarrow R$ satisfying the following conditions:*

1. *If g is unate, then h is also unate.*
2. *If g is ϵ -far from unate, then h is also ϵ -far from unate.*
3. *For all $x \in \{0, 1\}^d$, we can find $h(x)$ with one query to g .*

Proof. We use padding technique to construct h from g . Let $h(x, y) = g(x)$ where $x \in \{0, 1\}^m$ and $y \in \{0, 1\}^{d-m}$. The first and third conditions can be directly deduced from the definition of h . For the second condition, suppose, for the sake of contradiction that h is ϵ -close from unate. Then, there exists a unate function \tilde{h} , such that $\Pr_{x,y}[\tilde{h}(x, y) \neq h(x, y)] < \epsilon$. By an averaging argument, we can find a $\tilde{y} \in \{0, 1\}^{d-m}$ where, $\Pr_x[\tilde{h}(x, \tilde{y}) \neq h(x, \tilde{y})] < \epsilon$. Now, let $\tilde{g}(x) = \tilde{h}(x, \tilde{y})$ for all $x \in \{0, 1\}^m$. Thus, \tilde{g} is unate and $\Pr_x[\tilde{g}(x) \neq g(x)] < \epsilon$, contradicting the assumption that g is ϵ -far from unate. \square

Now we can construct a tester for a given input $g : \{0, 1\}^m \rightarrow R$. Suppose, h is the function that satisfies Claim 3.7. We run the unateness tester on h and output its answer. By Claim 3.7, correctness of this tester follows from the correctness of the tester for h . Since in Theorem 3.4 we proved that the unateness tester for g uses $\Omega(m)$ queries and Claim 3.7 condition 3 holds, the tester for h uses the same number of queries. This means testing a function $f : \{0, 1\}^d \rightarrow R$ for unateness, where $|R| = o(\sqrt{d})$, requires $\Omega(m) = \Omega(|R|^2)$ queries. \square

Theorem 3.1 follows from Theorems 3.2, 3.4 and 3.6. \square

3.2 Lower Bound for Boolean Functions

In this section, we prove a nonadaptive, 1-sided error lower bound for testing unateness of Boolean functions over the hypercube domain. This lower bound technique is similar to the one used for proving the lower bound for monotonicity by Fischer et al. [9].

Theorem 3.8. *Every nonadaptive 1-sided error tester for unateness of functions of the form $f : \{0, 1\}^d \rightarrow \{0, 1\}$ must make $\Omega(\sqrt{d})$ queries.*

The proof of the theorem is by the application of Yao's minimax principle. A 1-sided error tester must reject only if it finds a violation, otherwise it must accept. We define a distribution over a set of "hard" functions and prove that for this hard distribution, a tester with query complexity q detects a violation with probability at most $O(q/\sqrt{d})$.

For $x \in \{0, 1\}^d$, define $|x|$ as the number of bits that are equal to 1. The set of hard functions is defined as follows: for every $i \in [\frac{d}{2}]$, define $f_i : \{0, 1\}^d \rightarrow \{0, 1\}$ as

$$f_i(x_1, \dots, x_d) = \begin{cases} 1 & \text{if } |x| > d/2 + \sqrt{d} \\ 0 & \text{if } |x| < d/2 - \sqrt{d} \\ x_i \oplus x_{i+\frac{d}{2}} & \text{otherwise} \end{cases}$$

Claim 3.9. *For every $i \in [\frac{d}{2}]$, f_i is ϵ -far from unate for some constant ϵ such that $0 < \epsilon \leq \frac{1}{4}$.*

Proof. For a fixed $i \in [\frac{d}{2}]$, group together all points that differ only in the i^{th} and $(i + \frac{d}{2})^{\text{th}}$ coordinate. There are 4 points in each group as illustrated in Figure 3.1. Overall, there are 2^{d-2} such independent groups.

It can be seen that a constant fraction of such groups violate unateness of f_i (by its definition). In each such group, the value of f_i on at least 1 point has to be changed (or "repaired") to make it unate. Thus, overall, a constant fraction of points have to be changed to make the function unate. Hence, f_i is ϵ -far from unate for some constant ϵ . \square

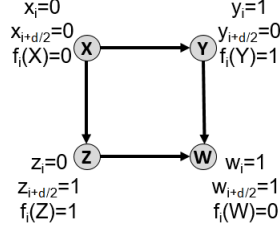


Figure 3.1: Illustration of “hard” function

Proof of Theorem 3.8. Let Q be the queried set of points from $\{0, 1\}^d$ of size q . By the definition of 1-sided error tester, we need to sample at least one violated, comparable pair of points to reject the function (A pair of points (x, y) is *comparable* if either $x \prec y$ or $y \prec x$). So, the tester can detect a violated pair for f_i if there exists two queries u, v such that they are comparable and differ in either i^{th} or $(i + \frac{d}{2})^{\text{th}}$ coordinate. Construct an undirected graph with vertex set Q and edge set $\{(x, y) | x \text{ and } y \text{ are comparable}, x, y \in Q\}$. Consider a spanning forest of this graph. If there exists a violation by function f_i in pair (u, v) they should be in the same tree. Moreover, there should exist an adjacent pair of vertices in the path between u and v which differ in the i^{th} bit. Therefore, the total number of functions f_i for which the query set could find the violation is at most the maximum number of edges in the spanning forest (which is at most $q - 1$) times the maximum possible distance between two adjacent vertices (which is at most $2\sqrt{d}$ with respect to our function because all violations are for $x \in \{0, 1\}^d$ such that $d/2 - \sqrt{d} \leq |x| \leq d/2 + \sqrt{d}$). So at most $O(q\sqrt{d})$ functions can be detected for violation. We have $\frac{d}{2}$ different functions. Hence, the number of queries required is at least $\Omega(\sqrt{d})$. \square

4 Open Questions

Our bounds for testing unateness are nearly tight for real-valued functions. However, there is a gap between the upper and lower bounds for Boolean functions. The first open question is to close this gap. In this paper, we consider testing unateness of real-valued functions with respect to the Hamming distance. Can one design testers with respect to other metrics such as L_p -distances, as defined in [2]?

References

- [1] Aleksandrs Belovs and Eric Blais. A polynomial lower bound for testing monotonicity. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 1021–1032, 2016.
- [2] Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. L_p -testing. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 164–173, 2014.
- [3] Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. *Computational Complexity*, 21(2):311–358, 2012.
- [4] Deeparnab Chakrabarty and C Seshadhri. Optimal bounds for monotonicity and lipschitz test-

- ing over hypercubes and hypergrids. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 419–428. ACM, 2013.
- [5] Deeparnab Chakrabarty and C. Seshadhri. A $o(n)$ monotonicity tester for boolean functions over the hypercube. In *Symposium on Theory of Computing Conference, STOC’13, Palo Alto, CA, USA, June 1-4, 2013*, pages 411–418, 2013.
 - [6] Xi Chen, Rocco A. Servedio, and Li-Yang Tan. New algorithms and lower bounds for monotonicity testing. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 286–295, 2014.
 - [7] Xi Chen, Anindya De, Rocco A. Servedio, and Li-Yang Tan. Boolean function monotonicity testing requires (almost) $n^{1/2}$ non-adaptive queries. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 519–528, 2015.
 - [8] Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. In *Randomization, Approximation, and Combinatorial Algorithms and Techniques, Third International Workshop on Randomization and Approximation Techniques in Computer Science, and Second International Workshop on Approximation Algorithms for Combinatorial Optimization Problems RANDOM-APPROX’99, Berkeley, CA, USA, August 8-11, 1999, Proceedings*, pages 97–108, 1999.
 - [9] Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 474–483, 2002.
 - [10] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Electronic Colloquium on Computational Complexity (ECCC)*, 3(57), 1996.
 - [11] Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, and Alex Samorodnitsky. Testing monotonicity. *Combinatorica*, 20(3):301–337, 2000.
 - [12] Subhash Khot and Igor Shinkar. An $\tilde{O}(n)$ queries adaptive tester for unateness. *RANDOM*, 2016.
 - [13] Subhash Khot, Dor Minzer, and Muli Safra. On monotonicity testing and boolean isoperimetric type theorems. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 52–58, 2015.
 - [14] Yue-Sun Kuo and WK Chou. Generating essential primes for a boolean function with multiple-valued inputs. In *Proceedings of the 23rd ACM/IEEE Design Automation Conference*, pages 193–199. IEEE Press, 1986.
 - [15] Alan Mishchenko. Introduction to zero-suppressed decision diagrams. *Applications of Zero-Suppressed Decision Diagrams*, 2014.

- [16] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.
- [17] Tsutomu Sasao. *Switching theory for logic synthesis*. Springer Science & Business Media, 2012.